

9 DTVCC Decoder Manufacturer Recommendations

The following are the recommendations for DTV Closed Captioning decoder implementation including DTV, SDTV and HDTV. These recommendations are directed to the least common denominator of all of the DTVCC features described in the previous sections. Although voluntary, these recommendations should be considered as requirements for a realistic minimal implementation of the DTVCC capabilities. These minimal recommendations provide a bridge from NTSC (EIA-608-A) captioning implementation to the eventual full-feature implementation of this DTVCC specification.

It should be emphasized that these minimum recommendations are not intended to, and should not, restrict caption providers from using the whole suite of DTVCC commands and their extensive capabilities. The following sections address the minimum recommendations that have been anticipated, but may not cover all conditions and manifestations. It is up to the manufacturer to consider all situations that are not explicitly presented herein.

NOTE--The section numbers in the following headings refer to the corresponding sections in the current DTVCC Specification, EIA-708-B, to which the minimum recommendations apply.

9.1 DTVCC Section 4.2 - Pre-Allocated Bandwidth

While the DTVCC Caption Channel provides a continuous 9600 bps bit stream within the DTVCC Transport Channel, the individual bandwidth allocated to any single service shall not exceed 25% of the total bandwidth averaged over any 1 second time interval. This limit permits a maximum, average captioning data rate of 300 Bps per service.

That is, decoders need only implement enough buffering and processing power to handle a maximum of 2400 bps for each service. In effect, when this limit is exceeded for a service, the input storage buffer allocated for the service will overflow and data not already buffered will be lost.

NOTE-- In contrast, the per-service limitation addressed above still provides a five-fold enhancement over the maximum possible NTSC Closed-Caption service data rate of 60 Bps.

9.2 DTVCC Section 6.1 - Services

Decoders should be capable of decoding and processing data for at least one (1) service. Decoders shall be capable of decoding and processing the Caption Service Directory data.

9.3 DTVCC Section 6.2 - Caption Channel Service Blocks

Decoders should be capable of decoding all Caption Channel Block Headers consisting of Standard Service Headers, Extended Service Block Headers, and Null Block headers. However, decoding of the data is required only for Standard Service Blocks (Service IDs ≤ 6), and then only if the characters for the corresponding language are supported.

Decoders should be able to display the directory for services 1 through 6. Service decoding and directory display for services numbered 7 or greater are optional.

9.4 DTVCC Section 7.1 - Code Space Organization

Decoders must support Code Space C0, G0, C1, and G1 in their entirety.

The following characters within code space G2 must be supported:

- transparent space (TSP)
- non-breaking transparent space ($NBTSP$)
- solid block (■)
- trademark symbol (TM)
- Latin-1 characters ($\text{Š}, \text{Œ}, \text{š}, \text{œ}, \text{Ÿ}$)

The substitutions in Table 17 are to be made if a decoder does not support the remaining G2 characters.

G2 Character	Substitute With
open single quote (‘), G2 char code 0x31	G0 single quote (‘), char code 0x27
close single quote (’), G2 char code 0x32	G0 single quote (‘), char code 0x27
open double quote (“), G2 char code 0x33	G0 double quote (“), char code 0x22
close double quote (”), G2 char code 0x34	G0 double quote (“), char code 0x22
bold bullet (*), G2 char code 0x35	G1 bullet (·), char code 0xB7
ellipsis(...), G2 char code 0x25	G0 underscore (_), char code 0x5F
one-eighth (¹ / ₈), G2 char code 0x76	G0 percent sign (%), char code 0x25
three-eighths (³ / ₈), G2 char code 0x77	G0 percent sign (%), char code 0x25
five-eighths (⁵ / ₈), G2 char code 0x78	G0 percent sign (%), char code 0x25
seven-eighths (⁷ / ₈), G2 char code 0x79	G0 percent sign (%), char code 0x25
vertical border (), G2 char code 0x7A	G0 stroke (I), char code 0x7C
upper-right border (), G2 char code 0x7B	G0 dash (-), char code 0x2D
lower-left border (), G2 char code 0x7C	G0 dash (-), char code 0x2D
horizontal border (—), G2 char code 0x7D	G0 dash (-), char code 0x2D
lower-right border (), G2 char code 0x7E	G0 dash (-), char code 0x2D
upper-left border (), G2 char code 0x7F	G0 dash (-), char code 0x2D

Table 17 G2 Character Substitution Table

Support for code spaces C2, C3, and G3 is optional.

All unsupported graphic symbols in the G3 code space are to be substituted with the G0 underscore character (_), char code 0x5F.

9.5 DTVCC Section 8.2 - Screen Coordinates

Table 18 specifies the screen coordinate resolutions and limits for anchor point positioning in 4:3 and 16:9 display formats, and the number of characters per row.

Screen Aspect Ratio	Maximum Anchor Position Resolution	Minimum Anchor Position Resolution	Maximum Displayed Rows	Maximum Characters per Row
4:3	75v x 160h	15v x 32h	4	32
16:9	75v x 210h	15v x 42h	4	42
other	75v x (5 x H)	15v x H*	4	*

Table 18 Screen Coordinate Resolutions & Limits

*H = 32 x (the width of the screen in relation to a 4:3 display). For example, the 16:9 format is 1/3 wider than a 4:3 display; thus, H = 32 * 4/3 = 42.667, or 42.

This means that the minimum grid resolution for a 4:3 aspect ratio instrument is 15 vertical positions x 32 horizontal positions. This minimum grid resolution for 16:9 ratio instrument is 15 vertical positions x 42 horizontal positions. These minimum grid sizes are to cover the entire safe-title area of the corresponding screen.

The minimum coordinates equate to a 1/5 reduction in the maximum horizontal and vertical grid resolution coordinates. Caption providers are to use the maximum coordinate system values when specifying anchor point positions. Decoders using the minimum resolution are to divide the provided horizontal and vertical screen coordinates by 5 to derive the equivalent minimum coordinates.

Any caption targeted for both 4:3 and 16:9 instruments is limited to 32 contiguous characters per row. If a caption is received by a 4:3 instrument that is targeted for a 16:9 display only, or requires a window width greater than 32

characters, then the caption may be completely disregarded by the decoder. 16:9 instruments should be able to process and display captions intended for 4:3 displays, providing all other minimum recommendations are met.

If the resulting size of any window is larger than the safe title area for the corresponding display's aspect ratio, then this window will be completely disregarded.

9.6 DTVCC Section 8.4 - Caption Windows

Decoders need to display no more than 4 rows of captions on the screen at any given time, regardless of the number of windows displayed. This implies that no more than 4 windows can be displayed at any given time (with each having only one caption row).

However, decoders should maintain storage to support a minimum total of 8 rows of captions. This storage is needed for the worst-case support of a displayed window with 4 rows of captioning and a non-displayed window which is buffering the incoming rows for the next 4-row caption.

As implied above, the maximum number of windows that may be displayed at any one time by a minimum decoder implementation is 4. If more than 4 windows are defined in the caption stream, the decoder may disregard the youngest and lowest priority window definition(s). Caption providers must be aware of this limitation, and either restrict the total number of windows used or accept that some windows will not be displayed.

9.7 DTVCC Section 8.4.2 - Window Priority

Decoders do not need to support overlapped windows. If a window overlaps another window, the overlapped window need not be displayed by the decoder. Decoders may support overlapped windows as an option.

9.8 DTVCC Section 8.4.6 - Window Size

At a minimum, decoders will assume that all windows have rows and columns "locked". This implies that if a decoder implements the optional SMALL pen-size, then word-"un"wrapping, when shrinking captions, need not be implemented. Also, if a decoder implements the optional LARGE pen size, then word wrapping (when enlarging captions) need not be implemented.

9.9 DTVCC Section 8.4.8 - Word Wrapping

Decoders may support word wrapping as an option.

9.10 DTVCC Section 8.4.9 - Window Text Painting

9.10.1 Justification

All decoders should implement "left", "right", and "center" caption-text justification. Implementation of "full" justification is optional. If "full" justification is not implemented, fully justified captions should be treated as though they are "left" justified.

For "left" justification, decoders should display any portion of a received row of text when it is received. For "center", "right", and "full" justification, decoders may display any portion of a received row of text when it is received, or may delay display of a received row of text until reception of a row completion indicator. A row completion indicator is defined as receipt of a CR, ETX or any other command, except SetPenColor, SetPenAttributes, or SetPenLocation where the pen relocation is within the same row.

Receipt of a character for a displayed row which already contains text with "center", "right" or "full" justification will cause the row to be cleared prior to the display of the newly received character and any subsequent characters. Receipt of a justification command which changes the last received justification for a given window will cause the window to be cleared.

9.10.2 Print Direction

At a minimum, decoders must support LEFT_TO_RIGHT printing.

9.10.3 Scroll Direction

At a minimum, decoders must support BOTTOM_TO_TOP scrolling.

For windows sharing the same horizontal scan lines on the display, scrolling may be disabled.

9.10.4 Scroll Rate

At a minimum, decoders must support the same recommended practices for scroll rate as is provided for NTSC closed-captioning.

9.10.5 Smooth Scrolling

At a minimum, decoders must support the same recommended practices for smooth scrolling as is provided for NTSC closed-captioning.

9.10.6 Display Effects

At a minimum, decoders must implement the “snap” window display effect. If the window “fade” and “wipe” effects are not implemented, then the decoder will “snap” all windows when they are to be displayed, and the “effect speed” parameter is ignored.

9.11 DTVCC Section 8.4.11 - Window Colors and Borders

At a minimum, decoders need only to implement borderless windows with solid, black backgrounds (i.e., border type = NONE, fill color = (0,0,0), fill opacity = SOLID), and borderless transparent windows (i.e., border type = NONE, fill opacity = TRANSPARENT).

9.12 DTVCC Section 8.4.12 - Predefined Window and Pen Styles

Predefined Window Style and Pen Style ID's may be provided in the DefineWindow command. At a minimum, decoders should implement Predefined Window Attribute Style 1 and Predefined Pen Attribute Style 1, as shown in Table 19 and Table 20.

Style ID #	Justify	Print Direction	Scroll Direction	Word Wrap	Display Effect	Effect Direction	Effect Speed	Fill Color	Fill Opacity	Border Type	Border Color	Usage
1	LEFT	LEFT -TO- RIGHT	BOTTOM -TO- TOP	NO	SNAP	n/a	n/a	(0,0,0) Black	SOLID	NONE	n/a	<i>NTSC Style PopUp Captions</i>
2	LEFT	LEFT -TO- RIGHT	BOTTOM -TO- TOP	NO	SNAP	n/a	n/a	n/a	TRANS- PARENT	NONE	n/a	<i>PopUp Captions w/o Black Background</i>
3	CNTR	LEFT -TO- RIGHT	BOTTOM -TO- TOP	NO	SNAP	n/a	n/a	(0,0,0) Black	SOLID	NONE	n/a	<i>NTSC Style Centered PopUp Captions</i>
4	LEFT	LEFT -TO- RIGHT	BOTTOM- TO- TOP	YES	SNAP	n/a	n/a	(0,0,0) Black	SOLID	NONE	n/a	<i>NTSC Style RollUp Captions</i>
5	LEFT	LEFT -TO- RIGHT	BOTTOM- TO- TOP	YES	SNAP	n/a	n/a	n/a	TRANS- PARENT	NONE	n/a	<i>RollUp Captions w/o Black Background</i>
6	CNTR	LEFT -TO- RIGHT	BOTTOM- TO- TOP	YES	SNAP	n/a	n/a	(0,0,0) Black	SOLID	NONE	n/a	<i>NTSC Style Centered RollUp Captions</i>
7	LEFT	TOP -TO- BOTTOM	RIGHT -TO- LEFT	NO	SNAP	n/a	n/a	(0,0,0) Black	SOLID	NONE	n/a	<i>Ticker Tape</i>

Table 19 Predefined Window Style ID's

Predefined Style ID	Pen Size	Font Style	Offset	Italics	Underline	Edge Type	Foregrnd Color	Foregrnd Opacity	Backgrnd Color	Backgrnd Opacity	Edge Color	Usage
1	STNDR	0	NORMAL	NO	NO	NONE	(2,2,2) White	SOLID	(0,0,0) Black	SOLID	n/a	<i>Default NTSC Style*</i>
2	STNDR	1	NORMAL	NO	NO	NONE	(2,2,2) White	SOLID	(0,0,0) Black	SOLID	n/a	<i>NTSC Style* Mono w/ Serif</i>
3	STNDR	2	NORMAL	NO	NO	NONE	(2,2,2) White	SOLID	(0,0,0) Black	SOLID	n/a	<i>NTSC Style* Prop w/ Serif</i>
4	STNDR	3	NORMAL	NO	NO	NONE	(2,2,2) White	SOLID	(0,0,0) Black	SOLID	n/a	<i>NTSC Style* Mono w/o Serif</i>
5	STNDR	4	NORMAL	NO	NO	NONE	(2,2,2) White	SOLID	(0,0,0) Black	SOLID	n/a	<i>NTSC Style* Prop w/o Serif</i>
6	STNDR	3	NORMAL	NO	NO	UNIFRM	(2,2,2) White	SOLID	n/a	TRANS-PARENT	(0,0,0) Black	<i>Mono w/o Serif, Bordered Text, No BG</i>
7	STNDR	4	NORMAL	NO	NO	UNIFRM	(2,2,2) White	SOLID	n/a	TRANS-PARENT	(0,0,0) Black	<i>Prop. w/o Serif, Bordered Text, No BG</i>

Table 20 Predefined Pen Style ID's

* "NTSC Style" - White Text on Black Background

9.13 DTVCC Section 8.5.1 - Pen Size

At a minimum, decoders must support the STANDARD pen size, with the implementation of the LARGE and SMALL pen sizes being optional.

The STANDARD pen size should be implemented such that the height of the tallest character in any implemented font is no taller than 1/15 of the height of the safe-title area, and the width of the widest character is no wider than 1/32 of the width of the safe-title area for 4:3 displays and 1/42 of the safe-title area width for 16:9 displays.

The LARGE pen size should be implemented such that the width of the widest character in any implemented font is no wider than 1/32 of the safe-title area for 16:9 displays. This recommendation allows for captions to grow to a LARGE pen size without having to reformat the caption since no caption will have more than 32 characters per row (see Section 8.4.6).

9.14 DTVCC Section 8.5.3 - Font Styles

Although a caption service provider may specify any one of 8 font styles using the **SetPenAttributes** command, decoders need only to implement a single font for caption text display.

Decoders that implement more than one font but do not support a font style specified in the **SetPenAttributes** command should instead display the caption text in the most similar font available. In decoders with only one font (i.e., font style 0, the default), all caption text, regardless of the specified font style, will be displayed in the default font.

In decoders with more than one but less than eight fonts, unsupported font styles should be displayed using an alternate font, giving precedence to the spacing attribute of the indicated font style, if possible. For example, if the specified but unsupported font style is "monospaced with serifs", the best substitute would be another monospaced font, and the second-best alternative would be a proportionally spaced font with serifs. If the Cursive font style is not supported, an acceptable substitution is an italicized version of an available font.

All supported font styles may be implemented in any typeface which the decoder manufacturer deems to be a readable rendition of the font style, and need not be in the exact typefaces given as examples in Section 8.5.3.

9.15 DTVCC Section 8.5.4 - Character Offsetting

Decoders need not to implement the character offsetting (i.e., subscript and superscript) pen attributes.

9.16 DTVCC Section 8.5.5 - Pen Styles

At a minimum, decoders must implement normal, italic, and underline pen styles.

9.17 DTVCC Section 8.5.6 - Foreground Color and Opacity

At a minimum, decoders must implement solid and flashing character foreground type attributes.

At a minimum, decoders must implement the following character foreground colors: white, black, red, green, blue, yellow, magenta and cyan.

9.18 DTVCC Section 8.5.7 - Background Color and Opacity

Decoders need only implement solid black character backgrounds. It is recommended that this background is extended beyond the character foreground to a degree that the foreground is separated from the underlying video by a sufficient number of background pixels to insure the foreground is separated from the background.

9.19 DTVCC Section 8.5.8 - Character Edges

Decoders need not to implement separate character edge color, opacity, and type attribute control. In this case, there is no separately controlled edge surrounding the body of characters.

9.20 DTVCC Section 8.8 - Color Representation

At a minimum, decoders must support the 8 colors described in Table 21.

Color	Red	Green	Blue
Black	0	0	0
White	2	2	2
Red	2	0	0
Green	0	2	0
Blue	0	0	2
Yellow	2	2	0
Magenta	2	0	2
Cyan	0	2	2

Table 21 Minimum Color List Table

When a decoder supporting this Minimum Color List receives an RGB value not in the list, it will map the received value to one of the values in the list via the following algorithm:

- All one (1) values are to be changed to 0
- All two (2) values are to remain unchanged
- All three (3) values are to be changed to 2

For example, the RGB value (1,2,3) will be mapped to (0,2,2), (3,3,3) will be mapped to (2,2,2) and (1,1,1) will be mapped to (0,0,0).

Table 22 is an alternative minimum color list table supporting 22 colors.

Color	Red	Green	Blue
Black	0	0	0
Gray	1	1	1
White	2	2	2
Bright White	3	3	3
Dark Red	1	0	0
Red	2	0	0
Bright Red	3	0	0
Dark Green	0	1	0
Green	0	2	0
Bright Green	0	3	0
Dark Blue	0	0	1
Blue	0	0	2
Bright Blue	0	0	3
Dark Yellow	1	1	0
Yellow	2	2	0
Bright Yellow	3	3	0
Dark Magenta	1	0	1
Magenta	2	0	2
Bright Magenta	3	0	3
Dark Cyan	0	1	1
Cyan	0	2	2
Bright Cyan	0	3	3

Table 22 Alternative Minimum Color List Table

When a decoder supporting the Alternative Minimum Color List in Table 22 receives an RGB value not in the list (i.e., an RGB value whose non-zero elements are not the same value), it will map the received value to one of the values in the list via the following algorithm:

- For RGB values with all elements non-zero and different - e.g., (1,2,3), (3,2,1), and (2,1,3), the 1 value will be changed to 0, the 2 value will remain unchanged, and the 3 value will be changed to 2.
- For RGB values with all elements non-zero and with two common elements - e.g. (3,1,3), (2,1,2), and (2,2,3), if the common elements are 3 and the uncommon one is 1, then the 1 elements is changed to 0; e.g. (3,1,3) -> (3,0,3). If the common elements are 1 and the uncommon element is 3, then the 1 elements are changed to 0, and the 3 element is changed to 2; e.g. (1,3,1) -> (0,2,0). In all other cases, the uncommon element is changed to the common value; e.g., (2,2,3) -> (2,2,2), (1,2,1) -> (1,1,1), and (3,2,3) -> (3,3,3).

All decoders not supporting either one of the two color lists described above, must support the full 64 possible RGB color value combinations.

9.21 Character Rendition Considerations

In NTSC Closed Captioning, decoders were required to insert leading and trailing spaces on each caption row. There were two reasons for this requirement:

- to provide a buffer so that the first and last characters of a caption row do not fall outside the safe title area, and

2. to provide a black border on each side of a character so that the "white" leading pixels of the first character on a row and the trailing "white" pixels of the last character on a row do not bleed into the underlying video.

Since caption windows are required to reside in the safe title area of the DTV screen, reason number 1 (above) is not applicable to DTVCC captions.

The attributes available in the **SetPenAttributes** command for character rendition (e.g., character background and edge attributes) provide unlimited flexibility to the caption provider when describing caption text in an ideal decoder implementation. However, manufacturers need only implement a minimum of pen attributes and font styles. Thus it is recommended that no matter what the level of implementation, decoder manufacturers should take into account the readability of all caption text against a variety of all video backgrounds, and should implement some automatic character delineation when the individual control of character foreground, background and edge is not supported; and when only a minimum number of font styles are implemented.

9.22 DTVCC Section 8.9 - Service Synchronization

Service Input Buffers must be at least 128 bytes in size. Caption providers must keep this lower limit in mind when following Delay commands with other commands and window text. In other words, no more than 128 bytes of DTVCC commands and text should be transmitted (encoded) before a pending Delay command's delay interval expires.

9.23 DTV to NTSC Transcoders

It is anticipated that receiver (decoder) manufacturers will develop devices (e.g., settop boxes) which process an DTV stream and transcode it for display on NTSC monitors. The DTVCC command set is not necessarily transcodable to NTSC captions; i.e., there are DTVCC captions which have no NTSC equivalent.

Although receiver manufacturers are free to attempt an automatic transcode of the captions, there is no guarantee that the captions will appear as the caption provider intended. Caption providers apply many techniques to make the captions easy to read and as unobtrusive as possible over the underlying video. To maintain caption quality during an automated transcode process, a set of conversion rules would have to be defined which cover all possible window, pen and text attribute combinations.

Therefore, a separate NTSC caption channel was added to the Picture User Data (see Section 4.3). This channel allows caption providers to encode dual caption streams within the same programming. NTSC captions are under the complete control of the caption provider; and thus, no automated transcoding of captions is necessary.

10 DTVCC Authoring and Encoding for Transmission

This section describes a DTV captioning "food chain". This is the path the captioning information takes from initial authoring intentions to being multiplexed into the ATSC emission bit stream. The content of this section is an example for information.

10.1 Caption Authoring and Encoding

High quality captioning starts with the creation of the captioning intentions. This is a high level, generally editable, representation of how and when the captions should appear when rendered on the consumer receiver. SMPTE 12M time code is generally used for synchronization with picture. The output of the initial authoring process is generally a computer file that contains a list of time codes and the intention as to what the receiver should render when the picture, with the corresponding time code, appears on the display device. This computer file is typically editable. The file may be stored on a hard disc or floppy disc, and distributed by either computer networking techniques, or via floppy net. This process is illustrated in Figure 19.

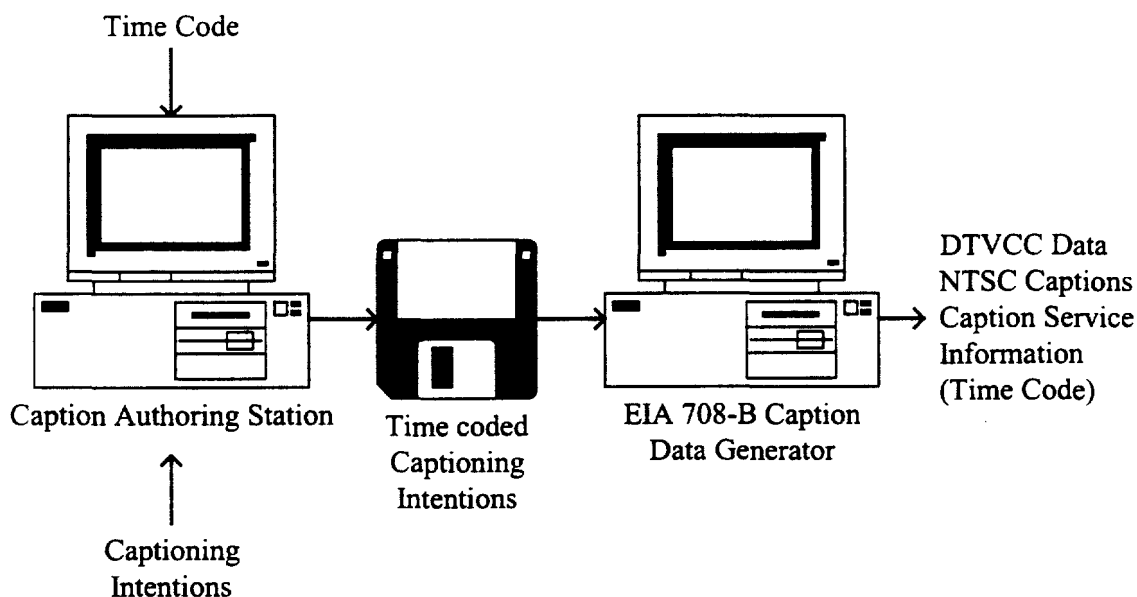


Figure 19 Caption Authoring and Encoding into Caption Channel Packets

The caption intentions are rendered into variable length *DTV Caption Channel Packets* (see Section 5). This rendering process is performed by an EIA-708-B caption data generator. This generator may also create a EIA-608-A (NTSC) captioning stream. This rendering process should consider the latency of the caption decoding process in the receiver, and thus should generate the packets pre-timed for transmission slightly early. Besides rendering the intentions into DTV Caption Channel Packets, the generator should create the caption service information that will be used to create the caption service descriptors that are carried in both the MPEG-2 PMT and in the ATSC PSIP EIT.

When delivered to the consumer receiver by the ATSC DTV system, the NTSC and DTV caption data is carried in the MPEG-2 picture user data, where a fixed number of bytes is allocated for each frame. The allocation provides 9,600 bits per second of data capacity. The number of bytes carried with each picture frame is dependent on the picture frame rate (see table 3 in section 4.4.2). In the case of 29.97 or 30 fps pictures, there are 4 bytes of NTSC

caption data and 36 bytes of DTV caption data carried per frame. The data rate of DTV Caption Channel Packets is equal to or lower than 9600 bps and so some zero padding is generally required. After the variable length DTV Caption Channel Packets are formed, it is necessary to parse these packets into the fixed length blocks that will be included with individual picture frames in the final emission multiplex. Many of these blocks will include zero padding so that the occupied bit rate is padded out to the full 9600 bps rate. This process is illustrated in figure 10-2.

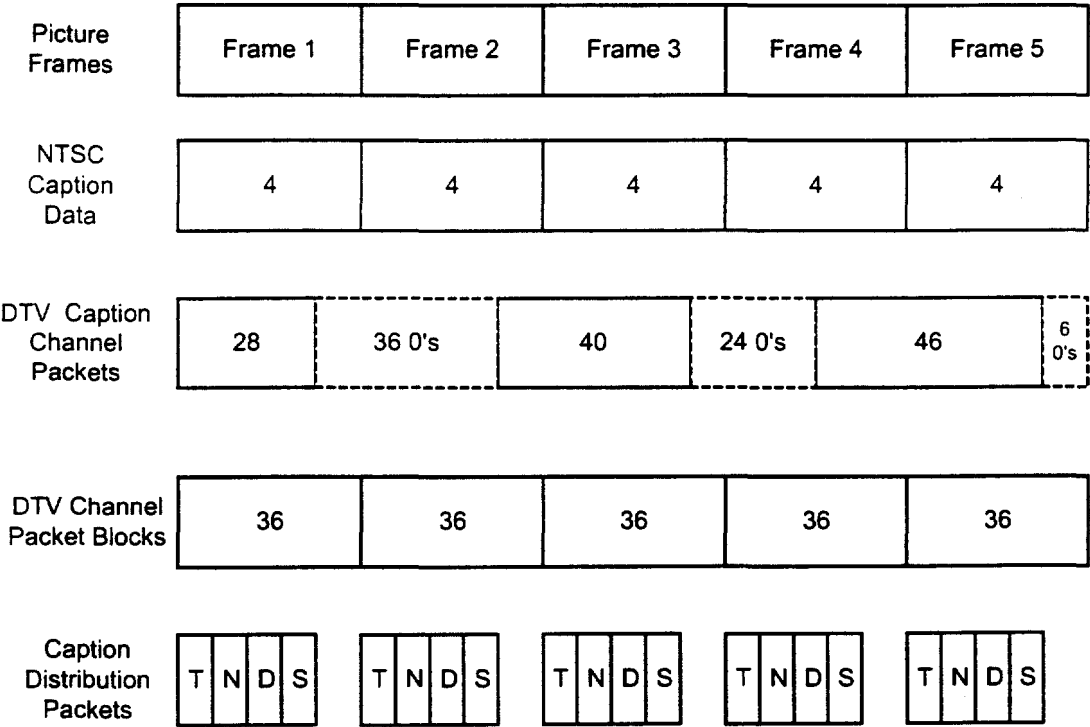


Figure 20 Relationship Between Caption Data and Picture Frames

Figure 20 illustrates five picture frame periods, at a 29.97/30 Hz frame rate. At this frame rate there are 4 bytes of NTSC caption data per frame. Three DTV Caption Channel Packets have been generated during the time of these 5 frames. These variable length packets have lengths of 28, 40, and 46 bytes. The DTV caption data is then formed into fixed length DTV Channel Packet Blocks that will be included in the corresponding MPEG-2 picture user data (see section 4) area. In this 29.97/30 Hz frame rate example, these blocks have a uniform length of 36 bytes. The first of these blocks contain the 28 bytes of the first DTV Caption Channel Packet plus 8 bytes of zero padding. The second block contains 28 bytes of zero padding followed by the first 8 bytes of the second DTV Caption Channel Packet, and so on. The result of this process is 36 bytes of DTV captioning data per picture frame. The final line in this figure shows the formation of all of the captioning data into *Caption Distribution Packets* that will be described in Section 11. These packets have a 1:1 relationship to video frames, and can include time code (T), the NTSC caption data (N), the blocked and zero padded DTV caption data (D), and the caption service information (S).

10.2 Monitoring Captions

Caption data may be monitored at various points during the distribution chain. To monitor the captions as they will be displayed on a consumer receiver, and to evaluate video/caption synchronization, it is necessary to decode the caption data after it has been rendered into a form (such as the Caption Distribution Packet) where there is an association of sets of caption data bytes with particular video frames.

10.3 Encoder Interfacing

DTVCC and NTSC captions are multiplexed into the Picture User Data in the MPEG-2 video elementary stream. This multiplexing is generally done within the MPEG-2 video encoder and so the caption data must be delivered to this encoder. A Caption Service Descriptor is included in the PMT and in the EIT and so caption service information must be delivered to the PSIP generator, and to the MPEG-2 PMT generator. (It is possible that both PMT and EIT would be created by a common functional block.) It is likely that all of the caption information will arrive at the encoding system on a single interface connection, and be distributed to the various destinations by looping the interface connection to multiple functional blocks. See Figure 21.

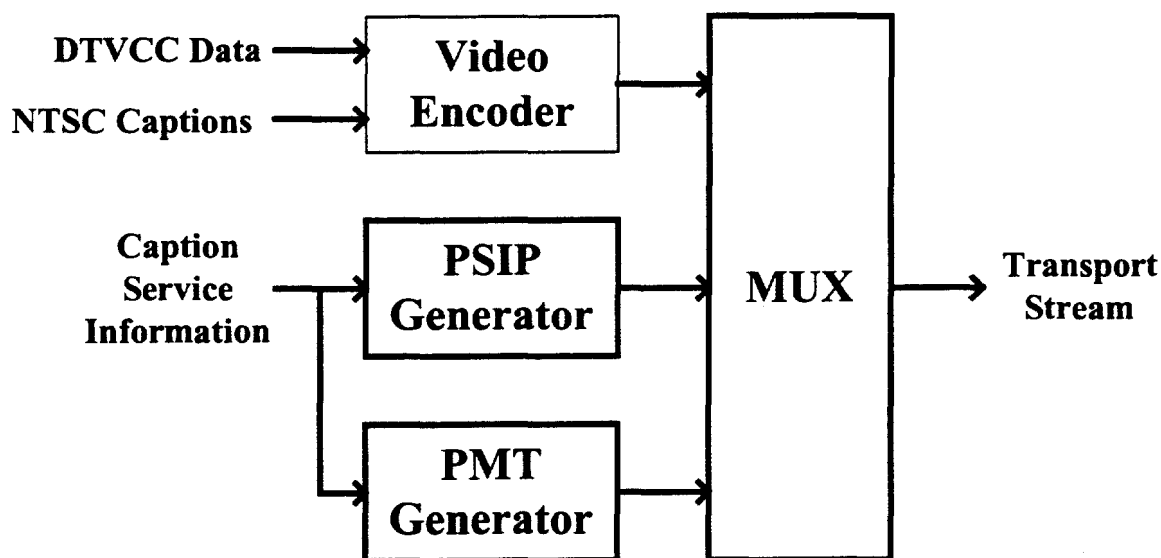


Figure 21 Interface of Caption Data to ATSC Emission Encoding Equipment

Several types of interfaces for the caption data may exist. These include a serial data format carried on an asynchronous RS232 like interface, embedding into an AES3 (digital audio interface) data stream, or embedding into SMPTE 259M or 292M serial digital video streams. For purposes of interoperability it is useful if there is as much commonality as possible to the data format on these different types of interfaces. The Caption Distribution Packet (CDP) described in Section 11 can carry DTVCC data, NTSC captions, caption service information, and time code.

EIA-708-B

Section 11 defines an asynchronous serial interface to carry the CDP. SMPTE is standardizing carriage of the CDP over AES3, 259M, and 292M interfaces.

11 DTV Closed Captioning Content Package

This section describes how to create a closed captioning content package, designated the Caption Distribution Packet (CDP), which may contain time code, EIA-708-B closed captioning data, and ATSC closed caption descriptor information. The CDP is suitable for transport over a variety of streaming interfaces. The details of one specific serial interface are defined.

11.1 Introduction

The process of creating and delivering closed captions for ATSC DTV involves authoring captions into a representation which represents the frame accurate captioning intentions (e.g. SAMI high level representation), rendering the intentions into the EIA-708-B caption syntax, transport of this EIA-708-B data via storage and/or streaming media to the point of emission, and then packaging the EIA-708-B data into the MPEG-2 picture user data as specified in ATSC A/53. During this process the caption data must be kept properly synchronized to the picture and sound. It is also necessary to create, transport, and include the caption service information in order to create the Caption Service Descriptor that is carried in the PMT and EIT tables in the MPEG-2 emission transport stream. During the distribution chain, the EIA 708-B captioning data may be rendered and displayed for quality control purposes.

This section describes how to create a captioning content package, consisting of a defined sequence of bytes, which may carry the following:

- time code;
- EIA-708-B caption;
- EIA-608-A NTSC caption data;
- caption service information to form the caption service descriptor; and/or
- sequence counts to detect discontinuities in the stream of caption data packets.

The CDP may be employed in various types of transport of captioning data, including file formats, RS232, TIA/EIA-574, AES3, SMPTE 259M, SMPTE 292M, etc. The CDP may be further encapsulated into a SMPTE defined Key-Length_Value (KLV) construct. Details on the KLV construct may be obtained from SMPTE.

11.1.1 Frame Rates

The transport of closed captioning information over the DTV emission system involves packaging the CC data into the MPEG-2 picture user data area. In order to do this the captioning data must be packaged at the same frame rate as that used by the video encoder. Rendering the captioning data into a particular frame rate is done prior to or during creation of the CDP. If the video encoder encodes at a frame rate that differs from the frame rate of the CDP, the captioning data must be re-framed. This can be done by the video encoder, by a caption data server, or by other equipment upstream of the video encoder. However, if the video encoder is responsible for the determination of the encoded picture frame rate, the re-framing should be done in the video encoder or in a captioning server tightly coupled (with 2-way communication) to the video encoder.

From the point of view of captioning, frame rates which differ by 0.1% may be considered identical. For example, if picture and captions are rendered at a 30 Hz frame rate, they may both be played at 29.97 Hz without any reframing of the caption data. As long as captions are delivered at the same rate as the picture, and this rate does not change by more than 0.1 %, no reframing is needed. Also, captions may be generated at a 25 or nominal 30 Hz rate, and used with 50 or nominal 60 Hz video. In this case, a video encoder should place the first half of the caption data in the first picture user data area, and place the second half of the caption data in the second picture user data area.

11.1.2 Time Code

The CDP may carry a time code which may be derived from SMPTE 12M VITC or LTC. Carriage of a time code provides an important tool to allow captions to be kept properly synchronized with pictures. The picture, sound, and caption data elements may flow through differing paths to the emission encoding and multiplexing equipment. The

inclusion of time code within each type of element makes it possible for the final multiplexer to provide buffering in order to deliver a properly synchronized multiplex of elements to the final receiver.

11.1.3 Caption Data

The caption data contained in the CDP is fully framed and formatted for direct inclusion within the ATSC video elementary stream picture user data, as defined in ATSC A/53, sections 5.2.2 and 5.2.3.

11.1.4 Caption Service Information

The caption service information in the CDP is fully framed and formatted for direct inclusion within the caption service descriptor as defined in ATSC A/65, section 6.7.3, table 6.17.

To reduce the data rate to carry the CDP stream (see section 11.1.5 below), the caption service information may be spread over a sequence of CDP packets. The receiver of the CDP stream must be able to collect service information from a sequence of CDP packets, and must be able to detect when the service information has changed.

Two general types of change may be envisioned. The first is a controlled change, where the generator of the CDP stream can insert an explicit indication that service information has changed, or that a service has been added or dropped. The second is an uncontrolled change that could be caused by a switch from one CDP stream to another CDP stream. In the case of an unsupervised switch, there can be no controlled signaling of the change, yet the receiver of the CDP stream must be able to easily detect that a change has occurred. In the event that a CDP stream is switched, the switch could result in a stream that has an incomplete, wrong, or damaged caption descriptor that should be discarded (and not transmitted to a consumer DTV receiver). To provide the ability to detect stream switches, 16-bit sequence counts are included in the CDP header and footer. If the received sequence counts do not increment smoothly, a switch or error has occurred.

11.1.5 Interface data rates

The data rate required to convey a CDP stream is dependent on the frame rate of the stream and on the amount of service information included within each packet. The worst-case data rate would occur at a 60 Hz frame rate, with 16 services all fully described within each CDP packet. In this case, the CDP packet size would be 161 bytes, and require a transmission rate of 9,660 bytes per second, or 96,600 bps over a serial interface with 1 start bit and 1 stop bit. If the service information is limited to describing only 1 service per CDP, then the maximum data rate becomes 3,360 bytes per second, or 33.6 kbps over the serial interface. It is therefore practical to carry the CDP stream over an TIA/EIA-574 38.4 kbps serial interface, although in some cases it is necessary to limit the amount of service information included in each CDP packet.

11.2 CDP Detailed Specification

The CDP shall be as specified in Section 11.2.

11.2.1 General Construct

The general construct of the CDP shall be as defined in Table 23.

<u>Syntax</u>	<u>Comment</u>
cdp() {	Caption Distribution Packet (CDP)
cdp_header();	Required
time_code_section();	Optional
ccdata_section();	Optional
ccsvcinfo_section();	Optional
future_section_1();	If defined, place here
future_section_2();	If defined, place here
cdp_footer();	Required
}	

Table 23 CDP General Construct

The CDP shall contain one header section and one footer section. The CDP may contain one time code section, one cc data section, and one cc service information section. The CDP shall not contain more than one of any of these

sections. These sections shall be multiplexed in the order shown in Table 23. It is possible to extend the CDP to include additional sections. Any sections that are defined in the future shall be placed just prior to the `cdp_footer`. Any newly defined sections would begin with a unique identifier byte, and contain a length code. The syntax that a new section would follow is shown in Section 11.2.7. Equipment that receives the CDP shall skip over sections that begin with an unknown id byte, by means of the length code.

11.2.2 cdp_header

The CDP header is a required element, and shall be present in all CDP's. CDP header syntax shall be as indicated in Table 24. The length of the `cdp_header` is fixed at 6 bytes.

Syntax	Bits	Format	Comment
<code>cdp_header() {</code>			
<code>cdp_identifier</code>	16	uimbsf	0x9669
<code>cdp_length</code>	8	uimbsf	length of entire packet
<code>cdp_frame_rate</code>	4	uimbsf	frame rate of packets
Reserved	4	'1111'	
<code>time_code_present</code>	1	bit	'1' indicates time code section is included
<code>ccdata_present</code>	1	bit	'1' indicates cc data section is included
<code>svcinfo_present</code>	1	bit	'1' indicates svcinfo section is included
<code>svc_info_start</code>	1	bit	'1' indicates start of svc info data set
<code>svc_info_change</code>	1	bit	'1' indicates change in cc service information
<code>svc_info_complete</code>	1	bit	'1' indicates completion of svc info data set
<code>caption_service_active</code>	1	bit	'1' indicates caption service is active
Reserved	1	'1'	
<code>cdp_hdr_sequence_cntr</code>	16	uimbsf	
<code>}</code>			

Table 24 CDP Header Syntax

cdp_identifier – This is a 16-bit value set to 0x9669. All CDP packets begin with this value.

cdp_length – This 8-bit integer shall indicate the number of bytes of data in the entire CDP packet, from the first byte of the `CDP_identifier`, to the packet checksum, inclusive.

cdp_frame_rate – This field shall indicate the frame rate of the CDP stream. It shall be coded as indicated in Table 25. Also shown are the values of `cc_count` and the number of `cc_data` bytes that shall be included in each packet at each frame rate.

cdp_frame_rate	frame rate	cc_count	NTSC cc_data bytes	DTV cc_data bytes
0000	Forbidden			
0001	24000÷1001 (~23.976)	25	6	44
0010	24	25	6	44
0011	25	24	----	50
0100	30000÷1001 (~29.97)	20	4	36
0101	30	20	4	36
0110	50	12	----	25
0111	60000÷1001 (~59.94)	10	2	18
1000	60	10	2	18
...	Reserved			
1111	Reserved			

Table 25 CDP Frame Rate

NOTE--There is no practical difference between the pairs of frame rates which differ by 0.1%. Captions and pictures rendered at one rate may be played 0.1% fast or slow, with no impact on presentation, as the number of bytes per frame does not change.

time_code_present – This bit shall be set to '1' for CDP packets which include the time code section. Otherwise this bit shall be set to '0'.

cc_data_present – This bit shall be set to '1' for CDP packets which include the cc data section. Otherwise this bit shall be set to '0'.

svcinfo_present – This bit shall be set to '1' for CDP packets which include the service information section. Otherwise this bit shall be set to '0'.

svc_info_start – This bit shall be set to '1' to indicate that the current packet begins a complete set of service information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_start bit in the cc service information section.

svc_info_change – This bit shall be set to '1' during the packet which begins a complete set of service information to indicate that the service information in the following set of information has changed from the previously delivered set of information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_change bit in the cc service information section.

svc_info_complete – This bit shall be set to '1' to indicate that the current packet concludes a full set of service information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_complete bit in the cc service information section.

caption_active – This bit shall be set to '1' to indicate that the CDP stream is conveying an active caption service. This bit shall be set to '0' in the case that the CDP stream is not conveying an active caption service.

cdp_hdr_sequence_cntr – This is an unsigned 16-bit integer which shall be set to a value of 1 plus the value of CDP_hdr_sequence_cntr in the previous CDP. The value of this counter shall wrap from 65535 to 0. For the first CDP in a sequence of CDPs, dcp_hdr_sequence_cntr may be set to any 16-bit value.

11.2.3 time_code_section

The time code section is optional in a CDP. Time code syntax is indicated in Table 26. This section shall be composed of a section id byte and 4 bytes of time code information. The length of the time code section shall be 5 bytes. Inclusion of this section may help assure that synchronization between captions and pictures is maintained throughout the distribution chain and into the final emission transport stream.

Syntax	Bits	Format	Comment
time_code_section() {			
time_code_section_id	8	uimsbf	0x71
Reserved	2	'11'	
tc_10hrs	2	uimsbf	Tens of hours
tc_1hrs	4	uimsbf	Units of hours
Reserved	1	'1'	
tc_10min	3	uimsbf	Tens of minutes
tc_1min	4	uimsbf	Units of minutes
tc_field_flag	1	uimsbf	see text
tc_10sec	3	uimsbf	Tens of seconds
tc_1sec	4	uimsbf	Units of seconds
Reserved	1	'1'	
drop_frame_flag	1	uimsbf	Drop frame flag
tc_10fr	3	uimsbf	Tens of frames
tc_1fr	4	uimsbf	Units of frames
}			

Table 26 CDP Time Code Section Syntax

time_code_section_id – This 8-bit field shall have the value of 0x71.

tc_field_flag – For interlaced pictures, the value of this flag shall be '0' for interlace field 1, and shall be '1' for interlace field 2. In the case of frame rates equal to or greater than 50 Hz, the frame count shall be interpreted as follows. The frame count shall be doubled, and the tc_field_flag shall be interpreted as an adder to the indicated frame count. The frame count shall be interpreted as (2 * frame + flag). I.e. the frame:flag sequence shall be 0:0, 0:1, 1:0, 1:1, 2:0, 2:1, etc., and this frame:flag sequence shall be interpreted as progressive frame counts 0, 1, 2, 3, 4, 5, etc.

drop_frame_flag – This flag shall be set to '1' when the time code count is being drop-frame compensated.. When the count is not drop-frame compensated, this flag bit shall be set to '0'.

11.2.4 ccdata_section

The ccdata section should normally be present. If present, the ccdata section syntax shall be as indicated in Table 27.

This section need not be included if a CDP stream is intended to carry only caption service information to a PMT or PSIP generator. However, a CDP stream typically conveys both cc data and cc service information in parallel to both the emission encoder and to the PMT and PSIP generators, with each device extracting and using the appropriate information.

This section shall be composed of a section id byte, a count value cc_count, and cc_count groups of 3 bytes. The total length of this section is 2 + 3 * cc_count bytes. The value of cc_count shall be dependent on the frame rate that is indicated in the CDP_header.

The actual caption data is carried in the cc_data_1 and cc_data_2 fields. The value of cc_count is found in table 10-2, and provides sufficient space to carry both NTSC and DTV caption data. The DTV caption data may represent up

to 16 caption services. If either the NTSC or DTV caption data is not present, the space is still allocated and filled with null (0x0) values. The NTSC caption data shall come first in this section, followed by the DTV caption data.

Syntax	Bits	Format	Comment
ccdata_section() {			
ccdata_id	8	0x72	Indicates ccdata section
marker_bits	3	'111'	
cc_count	5	uimsbf	number of cc constructs in section
for (i = 0 ; i < cc_count ; i++)			
{			
marker_bits	5	'1111 1'	
cc_valid	1	bslbf	as defined in 4.4.1
cc_type	2	bslbf	as defined in 4.4.1
cc_data_1	8	bslbf	
cc_data_2	8	bslbf	
}			
}			

Table 27 CDP CC Data Section Syntax

ccdata_id – This 8-bit field shall have the value 0x72.

cc_count – This 5-bit field shall indicate the number of cc_[xxx] data byte triplets carried in this section, and shall have the value appropriate to the frame rate as indicated in Table 25.

cc_valid – This bit shall be as defined in section 4.4.1. This bit indicates whether the following two bytes of cc_data contain valid captioning data or zero padding.

cc_type – This 2-bit field shall be as defined in section 4.4.1. This field indicates whether the following two bytes of data represent NTSC field 1 or 2 captions, or DVC channel packet data or DTVCC channel packet start.

cc_data_1 – This byte shall be as defined in section 4.4.3.

cc_data_2 – This byte shall be as defined in section 4.4.3.

11.2.5 ccsvcinfor_section

The ccsvcinfor section carries information for the Caption Service Descriptor. This section shall be composed of a section id byte, indication of controlled changes in the service information, an indication of the number of services that are described in the current packet, and caption service information. The ccsvcinfor_section syntax shall be as described in Table 28.

The complete set of caption service information may describe 1 to 16 different caption services. A complete set of service information may be included in the current packet, or may be distributed over a number of packets. The total length of this section is $2 + 7 * \text{svc_count}$ bytes.

Syntax	Bits	Format	Comment
ccsvcininfo_section() {			
ccsvcininfo_id	8	0x73	Indicates ccsvcininfo section
marker_bit	1	'1'	
svc_info_start	1	bit	
svc_info_change	1	bit	
svc_info_complete	1	bit	
svc_count	4	uimsbf	number of svc constructs in section
for (i = 0 ; i < svc_count ; i++)			
{			
reserved	3	'111'	
caption_service_number	5	uimsbf	
svc_data_byte_1	8	bslbf	
svc_data_byte_2	8	bslbf	
svc_data_byte_3	8	bslbf	
svc_data_byte_4	8	bslbf	
svc_data_byte_5	8	bslbf	
svc_data_byte_6	8	bslbf	
}			
}			

Table 28 CC Service Information Syntax

svc_info_start – This bit shall be set to '1' to indicate that the current packet begins a complete set of service information. Otherwise this bit shall be set to '0'. This bit shall be duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_start bit in the CDP header section.

svc_info_change – This bit shall be set to '1' during the packet which begins a complete set of service information to indicate that the service information in the following set of information has changed from the previously delivered set of information. Otherwise, this bit shall be set to '0'. This bit is duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_change bit in the CDP header section.

svc_info_complete – This bit shall be set to '1' to indicate that the current packet concludes a full set of service information. Otherwise this bit shall be set to '0'. This bit is duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_complete bit in the CDP header section.

NOTE--If a single packet contains a complete set of service information, then both the svc_info_start and svc_info_end bits would be set to '1'.

svc_count – This 4-bit field shall be set to a value equal to the number of services which have service information included in this service information section.

caption_service_number – This 5-bit field carries the caption service number for the service described by the following 6 service data bytes. This field shall have a value of 0x00 when the service data applies to the line 21 (EIA-608-A) service, and shall have a value (between 0x01 – 0x10 inclusive) that matches the caption_service_number contained within svc_data_byte_4 when the service data applies to one of the DTVCC services. This field shall not have a value between 0x11 and 0x1f (inclusive).

svc_data_byte_n – These 6 bytes shall carry the caption service data for one service, encoded as described by the caption service descriptor loop in ATSC A/65, section 6.7.3, table 6.17.

11.2.5.1 Service information signalling

The `svc_info` start, change, and end bits allow the receiver of a CDP stream to build up a complete set of cc service information from multiple packets, and to follow changes in the captioning services which are intentionally introduced by the source of the CDP stream. These changes may include beginning and ending of one or more captioning services, or an alteration in a particular service.

When a service is terminated, the `svc_info_change` bit shall be set to '1' during the first packet containing the next full set of service information. The following full sets of service information shall not contain any service information for the terminated service. The caption service number of the terminated service shall not appear in any of the `svc_info` sections.

When a service is changed, the `svc_info_change` bit shall be set to '1' during the first packet containing the next full set of service information. The following full sets of service information shall contain service information representing the new service information for the changed service.

If a CDP stream is switched, there should be a discontinuity in the sequence counters. If the switch occurs between packets, the discontinuity will occur between the previous footer value and the following header value. If the switch occurs during a packet, the balance of the new packet may not be received correctly and there will not be a correct sequence number or checksum in the footer. If a receiver detects that a CDP stream switch has occurred, the receiver should assume all service information has changed and take the next full set of service information as current.

NOTE--In this case, the `svc_info_change` bit will not signal a change of service information. Uncontrolled switches of CDP streams may cause momentary glitches in the display of captions on receivers.

11.2.6 cdp_footer

The CDP footer shall be present in all CDP packets. The CDP footer syntax shall be as defined in Table 29. This section contains a section id byte, a sequence counter value, and a checksum. The sequence counter provides good detection ability as to whether a switch occurred during the transmission of the current CDP packet. The check sum provides a simple means of detecting most transmission errors. The total length of this section is 4 bytes.

Syntax	Bits	Format	Comment
<code>cdp_footer() {</code>			
<code>cdp_footer_id</code>	8	0x74	Indicates CDP footer section
<code>cdp_ftr_sequence_cntr</code>	16	uimsbf	
<code>packet_checksum</code>	8	uimsbf	
<code>}</code>			

Table 29 CDP Footer Syntax

cdp_ftr_sequence_cntr – This 16-bit unsigned integer shall be set to the same value as the `cdp_hdr_sequence_cntr`. Receivers may use the values of `cdp_hdr_sequence_cntr` and `cdp_ftr_sequence_cntr` to detect that the entire packet has been received.

packet_checksum – This 8-bit field shall contain the 8-bit value necessary to make the arithmetic sum of the entire packet (first byte of `cdp_identifier` to `packet_checksum`, inclusive) modulo 256 equal zero.

11.2.7 future_section()

It is possible to define new sections to be included in the CDP. Any newly defined sections shall follow the syntax defined in this clause and in Table 30. All equipment that can receive the CDP shall be capable of ignoring these new sections. The length value is provided so that decoders will know how many bytes of data to skip.

Syntax	Bits	Format	Comment
future_section() {			
future_section_id	8	uimsbf	Value in range 0x75-0xEF
Length	8	uimsbf	Number of bytes of data
for (i = 0 ; i < length ; i++)			
{			
new_data_byte(i)	8		New data content
}			
}			

Table 30 future_section syntax

future_section_id -- Sections defined in the future shall be specified to have a section id value in the range 0x75 to 0xEF, inclusive. Decoders shall be designed to skip over sections that have id values that are not understood.

11.3 Serial Interface

This section defines one specific serial interface which may convey a CDP stream.

11.3.1 Physical Interface

The physical interface for a CDP stream shall be an TIA/EIA-574 interface. The source of the CDP stream shall be DTE with a 9-pin "D" male connector, and the receiver of the CDP stream shall be DCE with a 9-pin "D" female connector.

Table 31 and Table 32 detail the pin connections for the CDP serial interface. The terminology under the "Signal Name" column reflects that used in TIA/EIA-574. The source of the CDP stream shall follow the settings shown in Table 11-7. The receiver of the CDP stream shall follow the settings shown in Table 11-8. The minimum implementation of this serial interface only requires two wires: serial data and ground. In the case where the other 7 lines are connected to drivers, the state of those lines is specified.

Pin	In or Out	Signal Name	Setting	Comments
1	Input	Receive Line Signal Detect	NA	NC or ignored
2	Input	Receive Data	NA	NC or ignored
3	Output	Transmit Data	Active Data	CDP data stream
4	Output	DTE Ready	NC or "ON"	If connected, set "ON"
5	---	Common	GND	Signal common
6	Input	DCE Ready	NA	NC or ignored
7	Output	RTS	NC or "ON"	If connected, set "ON"
8	Input	CTS	NA	NC or ignored
9	Input	Ring Indicator	NA	NC or ignored

Table 31 CDP Source Connector Pinout (DTE Male)

Pin	In or Out	Signal Name	Setting	Comments
1	Output	Receive Line Signal Detect	NC or "ON"	If connected, set "ON"
2	Output	Receive Data	NC or "ONE"	If connected, set to "ONE" (Mark)
3	Input	Transmit Data	Active Data	DCCCP data stream
4	Input	DTE Ready	NA	NC or ignored
5	---	Common	GND	Signal common
6	Output	DCE Ready	NC or "ON"	If connected, set to "ON"
7	Input	RTS	NA	NC or ignored
8	Output	CTS	NC or "ON"	If connected, set to "ON"
9	Output	Ring Indicator	NC or "OFF"	If connected, set to "OFF"

Table 32 CDP Receiver Connector Pinout (DCE Female)

NOTE--NC means not connected; NA means not applicable.

The serial interface shall support operation with the parameters indicated in Table 33. The recommended baud rate is 38,400, but, if necessary, the interface may also be operated at 57,600 or 115,200 baud.

Parameter	Setting
Baud Rate	38,400 (default); 57,600 (optional); 115,200 (optional)
Data Bits	8
Parity	None
Stop Bits	1
Start Bits	1

Table 33 TIA/EIA-574 Interface Parameters for CDP Stream

11.3.2 Operation of the CDP Serial Interface

This section describes the typical application of the CDP serial interface. Other applications are not precluded.

In a typical application, captioning intentions are captured in a high level representation and then rendered into EIA-708 captioning packets. SMPTE time code may be employed to provide means for synchronizing the captioning intentions to the picture. With knowledge of the picture frame rate and time code, the EIA-708 captioning packets may be formed into CDP packets, where one CDP packet corresponds to each picture frame. During real-time streaming of pictures over a video interface and the corresponding CDP packets over the CDP serial interface, the CDP packet for picture frame n should be presented to the serial interface during the time window between the beginning of picture frame n and the beginning of picture frame $n+1$. In an ideal application, both the CDP packets and the individual pictures will have timecodes, and the MPEG-2 picture encoder will rely on these time codes to establish synchronization between the MPEG-2 encoded pictures and the captioning data included in the user data space of those coded pictures.

When the CDP is conveyed by this serial interface, the CDP shall be preceded by four null bytes (0x00). These null bytes, plus the `cdp_identifier` (0x9669), form a unique 48-bit sync code that allows the serial receiver to synchronize to the CDP stream.

NOTE—These null bytes are not considered part of the CDP, and are not required when the CDP is carried by other interfaces.

BIBLIOGRAPHY

ATSC A/54, Guide to the Use of the ATSC Digital Television Standard (1995)

- Advanced Television Systems Committee (ATSC), 1750 K Street N.W., Suite 1200, Washington, DC 20006; Phone 202-828-3130; Fax 202-828-3131; Internet <http://www.atsc.org/standards.html>

Annex A (Informative)

Figure 22 illustrates one example of a decoder implementation for processing the PMT, EIT and User Data in the DTVCC Transport Stream. It shows the separate paths of the Service Descriptors in the PMT and EIT and the service data in the Picture User Data bits.

NOTE--Different block diagrams may exist for other implementations.

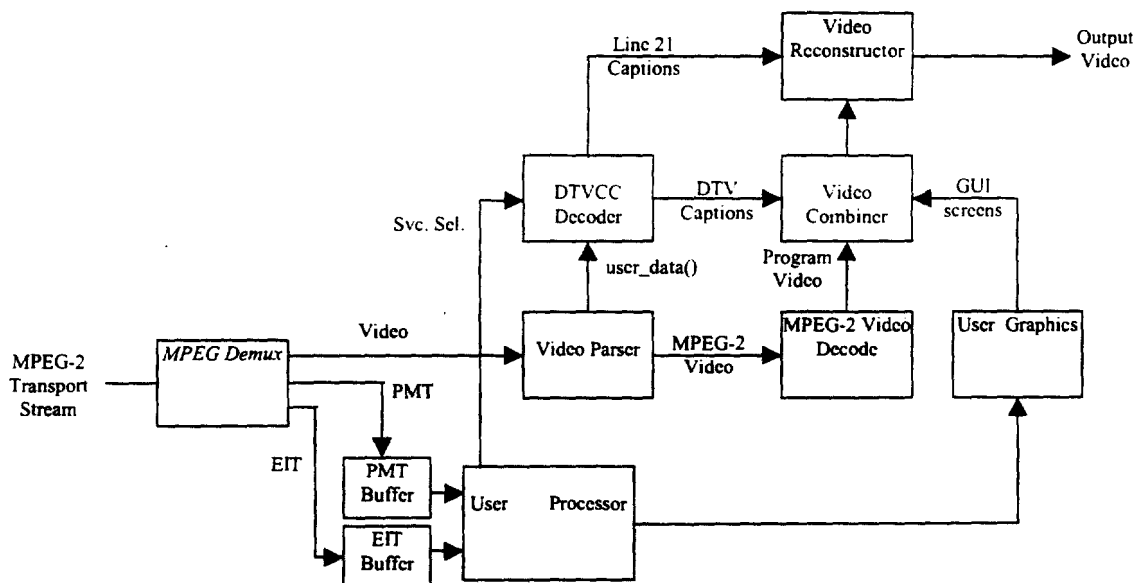


Figure 22 DTVCC Transport Stream Decoder

